

**METHOD OF SYNCHRONIZING AN INDEPENDENT DATA DEVICE OF A
WIRELESS DATA COMMUNICATIONS SYSTEM ON AN INCIDENT
PULSED SIGNAL OF THE ULTRA WIDE BAND TYPE, AND
CORRESPONDING INDEPENDENT DATA DEVICE**

Field of the Invention

[0001] The invention relates in general to sliding correlations as used in the communications field, and more particularly, in ultra wide band (UWB) radio technology for synchronizing a receiver on an incident signal. An example application of the invention is in the area of wireless personal area networks (WPAN) used to convey information over a relatively short distance among a relatively few participants.

Background of the Invention

[0002] An example wireless personal area network is a piconet, which is a wireless data communications system that allows a number of independent data devices to communicate with each other. A piconet is distinguished from other types of data networks in that communications are normally confined to a person or object that typically covers about 10 meters in all directions, and envelops the person or object whether stationary or in motion.

[0003] Ultra wide band radio technology departs from conventional narrow band radio and spread-spectrum technologies in that the bandwidth of the signal at -

10dB is typically greater than 20% of the center frequency. Instead of transmitting a continuous carrier wave modulated with information or with information combined with a spread code, which determines the bandwidth of the signal, a UWB radio transmits a series of very narrow impulses. For example, these impulses can take the form of a single cycle or monocycle, and have pulse widths less than 1 ns. These short time-domain impulses transformed into the frequency domain result in the ultra wide band spectrum of UWB radio technology.

[0004] In UWB radio technology, the information conveyed on the signal can be coded by pulse position modulation (PPM). In other words, information coding is performed by altering the timing of the individual pulses. More precisely, the series of impulses is transmitted at a repetition rate of up to several megahertz. Each pulse is transmitted within a window having a predetermined length (pulse repetition period), for example 50 ns. With respect to a nominal position, the pulse is in an early position or in a late position, which permits encoding of a 0 or a 1. It is also possible to encode more than two values by using more than two positions shifted with respect to the nominal position. It is also possible to superimpose a modulation of the BPSK type on this position modulation. More generally, any modulation can be used.

[0005] The IEEE 802.15.3 MAC standard defines the wireless medium access control specifications for high rate wireless personal area networks (WPAN) in narrow band radio technology. However, up to now, no document defines the specification of a wireless data

communication system, for example, a wireless personal area network using UWB radio technology, and more particularly, the synchronizing of two independent data devices of such a network communicating with each other with one of the devices acting as a coordinator of the network.

Summary of the Invention

[0006] An object of the invention is to provide an efficient way of implementing a synchronization phase, and more particularly but not exclusively, a coarse synchronization phase between a coordinator of the network and an independent data device. In particular, the synchronization requires very low complexity for digital data processing, and can be implemented in very low cost receivers having a reduced size receiving buffer.

[0007] This and other objects, advantages and features in accordance with the invention are provided by a method of synchronizing an independent data device of a wireless data communication system on an incident pulsed signal of the ultra wide band type received from a channel by the independent data device. The incident signal may contain a preamble including a preferably periodic training sequence having a series of pulses whose polarity and time shifts are defined by respective polarity code and time-hopping code.

[0008] The method may comprise a digital cross-correlation of the received signal with the training sequence. The cross-correlation step may include algebraically summing in accordance with the polarity code windows of the received signal, with the starting points of the windows being determined by the time-

hopping code. The method may also comprise a detection step for detecting the end point of the preamble from the result of the cross-correlation.

[0009] The size of the windows is the factor that constrains the buffer size used in the reception for storing the received samples. According to an embodiment of the invention, in which the training sequence is periodic and comprises replicas, each having a size of N samples and containing L pulses, and each window having a size of N samples, the digital cross-correlation is performed iteratively in a block-by-block fashion until a stop criterion is reached. The starting points of two consecutive blocks of correlation are separated by $2N$ samples.

[0010] For each iteration, the digital cross-correlation may comprise: a) initializing the content of an accumulation register capable of storing N data; b) taking a first group of N samples of the received signal starting from the starting point of the corresponding block increased by the time shift of the first pulse; c) multiplying the first group by the polarity of the first pulse; d) adding the resulting group of N samples to the content of the accumulation register; and repeating sub-steps b) to d) for all the L pulses.

[0011] However, in some cases, the requirements of the size of the receiving buffer could still be too large for very low cost receivers. Accordingly, when the size of the receiving buffer, i.e., the size of the window is smaller than N , it is particularly advantageous that the digital cross-correlation be performed iteratively in a block-by-block fashion until a stop criterion is reached. The computation of each

block is split into M slices which are computed by algebraically summing windows N/M samples long.

[0012] In such a case, the training sequence may comprise at least $M + 1$ replicas, with each replica having a size of N samples and containing L pulses, and with M being a sub-multiple of N greater than or equal to 2. The incident signals may carry information within a super frame or frame structure. Each super frame or frame containing the preamble includes at least $M+1$ synchronization slots corresponding respectively to the replicas of the training sequence. Depending on the signal-to-noise ratio, one synchronization slot may be sufficient to test a single slice.

[0013] In other words, in some cases, especially with a sufficiently high signal-to-noise ratio, each slice can be computed using one synchronization slot. In such a case, and according to an embodiment of the invention, the digital cross-correlation may be performed iteratively in a block-by-block fashion until a stop criterion is reached. The starting points of two consecutive blocks of correlation are separated by $N + N/M$ samples. For each iteration the digital cross-correlation step may comprise: a) initializing the content of an accumulation register capable of storing N/M data; b) taking a first group of N/M samples of the received signal starting from the starting point of the corresponding block increased by the time shift of the first pulse; c) multiplying the first group by the polarity of the first pulse; d) adding the resulting group of N/M samples to the content of the accumulation register; and repeating sub-steps b) to d) for all the L pulses. However, when the signal-to-noise ratio is not sufficiently high, each slice can be computed using

several adjacent synchronization slots belonging to several consecutive super frames.

[0014] Regardless of the case, the method according to the invention preferably comprises after each correlation iteration, a step of comparing the content of the accumulation register with a first predetermined threshold. The stop criterion may comprise the detection of at least one sample, called a peak, of the accumulation register having a value greater than the first predetermined threshold, or a predetermined maximum number of correlation iterations. The number of peaks that can be detected depends on the number of relevant paths (with large enough amplitude) of the channel.

[0015] According to an embodiment of the invention, the detection may comprise a first sub-step of detecting one replica of the training sequence. The first sub-step may comprise storing in memory the position of each peak in the accumulation register as well as its sign. Once a peak is detected, its position with respect to the correlation window is used to adjust the timing reference to the beginning of the next training sequence replica.

[0016] However, the first sub-step can successfully detect any of the replicas of the training sequence, while the object of the synchronization is to provide a receiver with the end point of the preamble, namely the index that identifies the last sample of the last replica of the training sequence. Thus, according to an embodiment of the invention, the preamble contains an additional flipped last replica of the training sequence and the detection step comprises a second sub-step, also called an alignment, including sequentially

scanning the next replicas until the flipped last one is found.

[0017] According to an embodiment of the invention, scanning a next replica may comprise performing a correlation between the next replica and the training sequence, comparing the correlation result with a second predetermined threshold, and if the absolute value of the correlation result exceeds the second threshold, using the sign of the correlation result and the sign of each detected peak to decide whether the next replica is the last one or if the scanning operation is to be performed again with the replica following the next replica.

[0018] Another aspect of the invention is directed to an independent data device for a wireless data communications system, for example of the WPAN type, and more particularly, of the piconet type. The independent data device may comprise reception means for receiving an incident pulsed signal of the ultra wide band type from a channel. The incident signal contains a preamble including a training sequence having a series of pulses whose polarity and time shifts are defined by respective polarity code and time-hopping code.

[0019] The independent data device may also comprise synchronization means comprising digital cross-correlation and detection means. The digital cross-correlation means may perform a cross-correlation of the received signal with the training sequence. The cross-correlation may include algebraically summing in accordance with the polarity code, windows of the received signal. The starting points of the windows are determined by the time-hopping code. The detection

means may detect the end point of the preamble from the result delivered by the cross-correlation means.

[0020] The training sequence may comprise replicas, each of which having a size of N samples and containing L pulses, each window having a size of N samples, and the digital cross-correlation means is adapted to perform the cross-correlation iteratively in a block-by-block fashion until a stop criterion is reached. The starting points of two consecutive blocks of correlation may be separated by $2N$ samples.

[0021] The cross-correlation means comprises an accumulation register capable of storing N data, and processing means. The processing means is adapted, for each iteration, to a) initialize the content of the accumulation register; b) take a first group of N samples of the received signal starting from the starting point of the corresponding block increased by the time shift of the first pulse; c) multiply the first group by the polarity of the first pulse; d) add the resulting group of N samples to the content of the accumulation register; and repeat sub-steps b) to d) for all the L pulses.

[0022] The training sequence may comprise at least $M+1$ replicas, each replica having a size of N samples and containing L pulses, with M being a sub-multiple of N greater than or equal to 2. The digital cross-correlation means is adapted to perform the cross-correlation step iteratively in a block-by-block fashion until a stop criterion is reached. The computation of each block is split into M slices which are computed by algebraically summing windows N/M samples long.

[0023] The incident signal carries information

within a super frame or a frame structure. Each super frame or frame containing the preamble includes at least $M+1$ synchronization slots corresponding respectively to the replicas of the training sequence. The cross-correlation means is adapted to compute each slice using one synchronization slot.

[0024] The digital cross-correlation means may be adapted to perform the cross-correlation step iteratively in a block-by-block fashion until a stop criterion is reached. The starting points of two consecutive blocks of correlation may be separated by $N + N/M$ samples. The cross-correlation means may comprise an accumulation register capable of storing N/M data, and processing means. The processing means may be adapted for each iteration, to: a) initialize the content of the accumulation register; b) take a first group of N/M samples of the received signal starting from the starting point of the corresponding block increased by the time shift of the first pulse; c) multiply the first group by the polarity of the first pulse; d) add the resulting group of N/M samples to the content of the accumulation register; and repeat sub-steps b) to d) for all the L pulses.

[0025] The incident signal carries information within a super frame or a frame structure. Each super frame or frame containing the preamble includes at least $M+1$ synchronization slots corresponding respectively to the replicas of the training sequence. The cross-correlation means is adapted to compute each slice using several adjacent synchronization slots belonging to several consecutive super frames.

[0026] The cross-correlation means may comprise comparison means for comparing after each correlation

iteration, the content of the accumulation register with a first predetermined threshold. The stop criterion may comprise the detection of at least one sample, called a peak, of the accumulation register having a value greater than the first predetermined threshold, or a predetermined maximum number of correlation iterations.

[0027] The detection means may comprise memory means and storing means for storing in the memory means the position of each peak in the accumulation register as well as its sign. The preamble may contain an additional flipped last replica of the training sequence, and the detection means may comprise scanning means for sequentially scanning the next replicas until the flipped last one is found.

[0028] The scanning means may comprise correlation means, comparison means and control means. The correlation means performs a correlation between the next replica and the training sequence. The comparison means compares the correlation result with a second predetermined threshold. For the control means, if the absolute value of the correlation result exceeds the second threshold, the sign of the correlation result and the sign of each detected peak are used to decide whether the next replica is the last one, or if the scanning operation must be performed again with the replica following the next replica.

Brief Description of the Drawings

[0029] Other advantages and features of the invention will appear on examining the detailed description of embodiments, these being in no way limiting and of the appended drawings in which;

[0030] Figure 1 shows diagrammatically an architecture of a wireless personal area network according to the invention;

[0031] Figure 2 shows diagrammatically an example of a super frame structure according to the invention;

[0032] Figure 3 shows diagrammatically an internal architecture of an independent data device according to the invention;

[0033] Figures 4 and 5 show diagrammatically in greater detail, portions of the independent data device illustrated in Figure 3;

[0034] Figures 6-8 relate to a first embodiment of a method according to the invention;

[0035] Figures 9-11 relate to a second embodiment of a method according to the invention;

[0036] Figures 12 and 13 relate more particularly to an alignment step of the method according to the invention; and

[0037] Figures 14 and 15 show diagrammatically two possible ways of computing signal slices in a slice-splitting correlation according to the invention.

Detailed Description of the Preferred Embodiments

[0038] Figure 1 shows a wireless communication system, such as a wireless personal area network. This wireless communication system network can be organized, for example, in a piconet PN fashion, which allows a number of independent data devices DEVi to communicate with each other. A WPAN can be composed of several piconets.

[0039] A piconet is distinguished from other types of data networks in that communications are normally confined to a person or object that typically covers

about 10 meters in all directions, and envelops the person or object whether stationary or in motion. A piconet is in contrast to local area networks (LAN), metropolitan area networks (MAN), and wide area networks (WAN) that cover a large geographic area, such as a single building or a campus or that would interconnect facilities in different parts of a country or the world.

[0040] The basic component of a piconet is a independent data device DEV. Such an independent data device may be, for example, a personal computer or the like. One independent data device DEV is required to assume the role of the piconet coordinator PNC. The piconet coordinator PNC provides the basic timing for the piconet with a beacon that is a part of a super frame, as will be explained in greater detail below. The PNC can communicate with the independent data devices DEV_i. Further, two independent data devices can communicate with each other directly, in a peer-to-peer manner, without relaying through the PNC.

[0041] Figure 2 depicts a super frame structure used in the present invention for the communication between the coordinator PNC and the data devices DEV, as well as for the communication between two independent data devices.

[0042] The incident signal received from a channel by an independent data device DEV carries information within a super frame structure. Each super frame SPF_i includes several frames FR_i respectively allocated to communications between specific pairs of independent data devices DEV.

[0043] Further, each super frame includes a head part, also called a beacon BC. The beacon contains a

preamble PRB including a periodic training sequence TS1 and a body including time of arrival indication TOA_j for each frame FR_j. Further, each frame FR_i includes also a preamble PRB containing a training sequence TS2 and a body portion BD containing the useful data.

[0044] As illustrated in Figure 3, an independent data device DEV basically comprises reception means RCM followed by a processor PR. More precisely, reception means comprises an antenna ANT followed by a conventional front end radio frequency stage. The processor comprises synchronization means SYM which includes digital cross-correlation means CRM and detection means DM. The internal architecture of the cross-correlation means CRM and the detection means DM will be described in greater detail below.

[0045] When an independent data device DEV is turned on, it will first search for an existing piconet in its neighborhood, and then synchronize to its coordinator PNC. This operation is named cell synchronization, or DEV-to-PNC synchronization.

[0046] The cell synchronization operation can be subdivided in three distinct phases: the coarse synchronization, the fine synchronization, and the clock synchronization. During the coarse synchronization, the independent data device DEV is looking for the training sequence TS1 sent by the coordinator PNC. The object of the digital coarse synchronization is to provide the independent data device with an index of the sample corresponding to the end of the beacon preamble.

[0047] During the fine synchronization, the independent data device synchronizes precisely to the beginning of the beacon body which contains, for

example, the different indications TOA_j to be sent by the coordinator to all independent data devices DEV of the piconet. The clock synchronization includes identifying an eventual drift between the coordinator's clock and the independent data device clock.

[0048] The different ways of implementing a coarse synchronization will now be described in greater detail. Generally speaking, during a first phase including a digital cross-correlation step, one of the replicas of the periodic training sequence TS1 will be detected. Then, during a second phase, the end of the beacon preamble will be actually detected.

[0049] In the beacon preamble a known training sequence TS1, also referred to as $s(t)$, is transmitted by the piconet coordinator (PNC). To perform the synchronization with the PNC, a device has to detect this signal in the presence of noise and interference, and from this have a timing reference for synchronization. The training signal $s(t)$ has the form of a pulsed UWB signal in Equation (1):

$$s(t) = \sum_{j=0}^{L-1} a_j p(t - jT_f - c_j T_c) \quad (1)$$

Indeed, $s(t)$ is a series of L pulses $p(t)$, nominally spaced by T_f (pulse repetition period: PRP) and whose polarity and time shifts (time-hopping) are defined by the sequences (or codes) a_j and c_j , respectively. T_c represents the granularity of the time shift (i.e., the minimum separation between two pulse positions inside a PRP). In another context, for a 2-PPM data modulation, T_c is the time duration between an early pulse and a

late pulse, as mentioned above with reference to a nominal position. It is useful to represent $s(t)$ as:

$$s(t) = p(t) * \sum_{j=0}^{L-1} a_j \delta(t - jT_f - c_j T_c) = p(t) * b(t) \quad (2)$$

where $*$ denotes the linear convolution operator and $b(t)$ includes the contributes of the codes a_j and c_j .

[0050] It is assumed that the received signal $r(t)$ corresponds to $s(t)$ delayed by a certain time τ plus a zero mean gaussian noise process $n(t)$ with power equal to σ^2 . It is known that the detection of the training sequence is based on the cross-correlation of $r(t)$ with the signal $s(t)$ and can be implemented for instance, with an analog matched filter.

[0051] The final goal of synchronization is to have an estimate $\hat{\tau}$ of the delay τ . However, in digital UWB radio technology, the synchronization means has access only to samples of the received signal. The correlation or matched filter has thus to be implemented digitally by using samples whose sampling frequency is determined by the system.

[0052] The signals are redefined as follows:

$$s_n = \sum_{j=0}^{L-1} a_j p_{n-c_j} = p_n * \sum_{j=0}^{L-1} a_j \delta_{n-c_j} = p_n * b_n \quad (3)$$

$$r_n = s_n + n_n \quad (4)$$

where p_n is the sampled version of $p(t)$. The sequences a_j and c_j are the polarity and time-hopping patterns used in the training sequence. The training sequence

comprises replicas, each of which having a size of N samples and containing L pulses. In other words, the training sequence is periodic with a period of N samples.

[0053] The output signal x_n is the sum of a signal term y_n and a noise term q_n :

$$x_n = r_n * c_{-n} = \sum_{j=0}^{L-1} a_j r_{n+d_j} = y_n + q_n \quad (5)$$

The signal term y_n is the following:

$$y_n = s_{n-m} * c_{-n} = p_n * c_n * \delta_{n-m} * c_{-n} = p_n * R_{cc}(n) * \delta_{n-m} \quad (6)$$

where $R_{cc}(n)$ is the autocorrelation function of c_n and δ_n is the Dirac delta function.

[0054] As illustrated in Figure 4, the cross-correlation means CRM comprise a receiving buffer RB. Theoretically, to correctly perform the block-by-block circular correlation, the processor PR has to be extremely fast because the result must be achieved and the receiving buffer RB emptied before the arrival of the next block of the received signal. The invention provides an implementation that avoids such a disadvantage, and the invention uses the periodicity properties of the signal, i.e., the training sequence.

[0055] In this case, the matched filtering can be performed in a circular way and the periodic autocorrelation function of b_n , $\tilde{R}_{bb}(n)$ appears in the expression of y_n . $\tilde{R}_{bb}(n)$ is periodic with a period of N samples and so the correlation process can be performed

in a block-by-block fashion, iteratively filtering blocks of N samples of the received signal.

[0056] In other words, the received signal r_n is split into blocks of N samples. At each iteration of the synchronization algorithm one block is received as input, a circular correlation with b_n is computed and the result is passed to a threshold device. More precisely, for the periodicity properties of the signal, a block circularly shifted of c_j samples has the same signal content of a block taken from the received signal, but starting c_j samples after. In this way the circular correlation can be computed by using different windows $w_n^{(i,j)}$ of the received signal, whose starting points are selected according to the time-shifts c_j , as depicted in Figure 6.

[0057] The i^{th} output block $x_n^{(i)}, n=0,1,..N-1$ is obtained as follows:

$$x_n^{(i)} = \sum_{j=0}^{L-1} a_j r_{n+2Ni+c_j} = \sum_{j=0}^{L-1} a_j w_n^{(i,j)}, \quad i = 0, 1, \dots \quad (7)$$

The term $2Ni + c_j$ corresponds to the starting times of the windows $w_n^{(i,j)}$ of the received signal. The starting points of consecutive blocks of correlation are separated by $2N$ samples, as depicted in Figure 7. The sliding correlation algorithm according to this embodiment requires very low complexity for digital data processing. As polarity signaling is used, the weights a_j are ± 1 , and therefore, the whole synchronization algorithm turns out to be the sum (or difference) of windows of the received signal, whose starting points are determined by the time-hopping code

used in the training sequence.

[0058] The flow chart of the sliding correlation according to this embodiment is also illustrated in Figure 8. In this figure, it is assumed that $n_i=0$ identifies the first available sample in the FIFO reception buffer RB (figure 4). Each iteration i of the digital cross-correlation phase is performed by processing means PRM (Figure 4). Further, the cross-correlation means CRM comprises an accumulation register ACR for storing the value x .

[0059] The cross-correlation CRM further comprises comparison means CMP for comparing the absolute value of x with a first predetermined threshold TH1. As illustrated in Figure 8, after the content of the accumulation register has been initialized to 0 (step 80) the i^{th} correlation iteration 81 is performed.

[0060] More precisely, this iteration comprises (step 810) taking N samples of the received signal starting from the starting point of the corresponding block increased by the time shift c_j of the first pulse.

[0061] Then, the first group is multiplied by the polarity a_j of the first pulse (step 811). The resulting group of N samples is added (step 812) to the content of said accumulation register, and these sub-steps are repeated for all the L pulses.

[0062] The object of the synchronization algorithm is to provide the receiver with the index $n_{\text{synchronro}}$ of the sample corresponding to the end of the beacon preamble. A part of this information is obtained by the correlation means, and more precisely, by the position and the sign of a peak detected after comparison with the threshold TH1 (step 82).

[0063] More precisely, if a peak is successfully

detected (the absolute value of a sample of the correlation output is above the threshold), its position n_{peak} and its sign α_{peak} are stored (step 83) by storing means STM into memory means MM (Figure 5). The sign of the peak holds information on the absolute phase of the received signal and is also needed to obtain $n_{synchro}$. In fact, n_{peak} is not sufficient to determine $n_{synchro}$. The training sequence has been assumed to be periodic with period N , and hence $n_{synchro}$ provides information with an ambiguity modulo N . It will be explained in greater detail below on how to solve this ambiguity modulo N .

[0064] In Figure 8, it has been assumed that the stop criterion of the outer loop was the detection of a peak. However, in some cases, the signal-to-noise ratio can be so low that no detection is possible. Another possibility is that no coordinator is transmitting, and consequently there is nothing to detect. Therefore, it is preferable that the stop criterion comprises also a condition on a maximum number of iterations.

[0065] Further, a single peak is likely to exceed the threshold TH1 only if a single path channel is assumed. In the case of a multipath channel, many peaks may exceed the threshold TH1. Here (Figure 9) and in the following, it is assumed that a single path channel is present.

[0066] The optimization of the algorithm according to the invention with multipath propagation will be treated in greater detail below. The main advantage of the embodiment which has been described is a very low complexity of the control needed for the processing of data. Nevertheless, the requirements of the size of the receiving buffer could still be too large for very low

cost receivers.

[0067] More precisely, with the sliding correlation algorithm described above, the i^{th} iteration (or block) of the full correlation is obtained as:

$$x_n^{(i)} = \sum_{j=0}^{L-1} a_j w_n^{(i,j)}, \quad n = 0, 1, \dots, N-1 \quad (8)$$

[0068] Namely, the block is the result of the sum of L weighted windows $w_n^{(i,j)}$ of the received signal. The size of such windows is the factor that constrains the minimum buffer size in the reception, which must be able to store N samples. As a numerical example, if the training sequence is $1 \mu\text{s}$ long, a sampling frequency equal to 20 GHz implies that the buffer in reception must be able to store $N = 20000$ samples.

[0069] However, according to one possible implementation suitable for very low cost receivers, only $N = 1280$ samples can be stored in the buffer, corresponding to a training sequence with duration 64 ns sampled at 20 GHz. Thus, it is particularly advantageous to reduce the complexity required by the sliding correlation algorithm detailed above.

[0070] The invention proposes accordingly in the embodiment which will be now described, a slice splitting correlation algorithm. With respect to how the sliding correlation is performed, it is clear that the n^{th} sample of the output block is the result of the sum of the n^{th} samples in the windows of the received signal that are summed to get the output block.

[0071] Indeed, nothing prevents the algorithm from just computing part of the whole block, for instance

the first half. The first $N/2$ samples of the output block are obtained by just considering the first halves of the windows used previously, and the same happens for the second half of the block. In other words, the computation of the i^{th} block can be split into two slices, which can be computed by adding weighted windows $N/2$ samples long. The following relationship holds between a block and the corresponding slices:

$$x_n^{(i,k)} = x_{n+k\frac{N}{2}}^{(i)}, \quad n = 0, 1, \dots, N/2-1, \quad k = 0, 1 \quad (9)$$

where $x_n^{(i,k)}$ denotes the k^{th} slice of the i^{th} block. From the received signal r_n , the slice $x_n^{(i,k)}$ can be obtained by combining Equation (8) with Equation (9):

$$x_n^{(i,k)} = \sum_{j=0}^{L-1} a_j r_{n+3iN+kN+k\frac{N}{2}+c_j} = \sum_{j=0}^{L-1} a_j w_n^{(i,k,j)}, \quad n = 0, 1, \dots, N/2-1, \quad k = 0, 1 \quad (10)$$

Figure 9 describes the relationship between the block $x_n^{(i)}$ and the slices $x_n^{(i,k)}$ and how these are obtained from the received signal.

[0072] The delay term $3iN + kN + kN/2 + c_j$ in Equation (10) describes the starting points of the windows to be summed and deserves a more detailed explanation. With the help of Figure 9, the meanings of all the contributions are clear. c_j is the time-shift given by the time-hopping code. $kN/2$ is needed to properly align the slices inside the block. kN is needed to prevent windows of the same block but of different slices from overlapping. $3iN$ is needed to

prevent windows of different blocks from overlapping.

[0073] The slice splitting procedure can be generalized with more than 2 slices. Generally speaking, a block of N samples can be split into M slices of N/M samples, where M is a sub-multiple of N . The slices are defined as follows:

$$x_n^{(i,k)} = x_{n+k\frac{N}{M}}^{(i)}, \quad n = 0, 1, \dots, N/M-1, \quad k = 0, 1, \dots, M-1 \quad (11)$$

And Equation (8) can be modified accordingly:

$$x_n^{(i,k)} = \sum_{j=0}^{L-1} a_j r_{n+[i(M+1)+k]N+k\frac{N}{M}+c_j} = \sum_{j=0}^{L-1} a_j w_n^{(i,k,j)} \quad (12)$$

[0074] As before, the delay term $i(M+1)N + kN + kN/M + c_j$ in Equation (12) describes the starting points of the windows $w_n^{(i,k,j)}$ to be summed: c_j is the time-shift given by the time-hopping code; kN/M is needed to properly align the slices inside the block; kN is needed to prevent windows of the same block but of different slices from overlapping; and $i(M+1)N$ is needed to prevent windows of different blocks from overlapping.

[0075] At this stage, a change of notation is useful: an index m is defined, which combines the index i (block of full correlation) and the index k (slice of the correlation):

$$m = (i-1) \cdot M + k, \quad i = 0, 1, \dots, \quad k = 0, 1, \dots, M-1 \quad (13)$$

And conversely

$$\begin{aligned} i &= \lfloor m/M \rfloor \\ k &= m \bmod M \end{aligned} \tag{14}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation.

[0076] As it is well known by those skilled in the art, the floor function applied to x gives the largest integer less than or equal to x . With this notation, the starting point of the windows is shifted exactly $N + N/M$ samples at every increment of m .

$$x_n^{(m)} = \sum_{j=0}^{L-1} a_j r_{n+m\left(N+\frac{N}{M}\right)+c_j} = \sum_{j=0}^{L-1} a_j r_{n+n_m+c_j} = \sum_{j=0}^{L-1} a_j w_n^{(m,j)} \tag{15}$$

$w_n^{(m,j)}$ is a window of N/M samples, starting at sample n_m of r_n .

[0077] It can be verified that an iteration of the full correlation ($m = 0, 1, \dots, M-1$) does not require more than $M + 1$ repetitions of the training sequence in the beacon preamble. While Figure 8 illustrates the flow chart of the sliding correlation according to the invention, Figure 10 illustrates the flow chart of the slice splitting correlation according to the invention.

[0078] Except for the number of samples in windows, the steps illustrated in Figure 10 are analogous to those illustrated in Figure 8, and will not be described in detail here. More precisely, in Figure 10, steps 100, 101, 102 1010, 1011 and 1012 are analogous to the corresponding steps 80, 81, 82, 810, 811 and 812 of Figure 8.

[0079] The slice splitting algorithm permits trade

off between the complexity of the receiver (in particular, the size of the buffer needed in reception) against the duration of the beacon preamble. In fact the overall duration of the beacon preamble, $N(M+1)$ samples depends linearly on the number of slices M while the length of the buffer N/M samples is inversely proportional to M .

[0080] As indicated above, either for the sliding correlation of the first embodiment, or the slice splitting correlation of the second embodiment, n_{peak} is affected by an ambiguity modulo N samples, deriving from the fact that the correlation can successfully detect any of the replicas of the training sequence. As an example, Figure 11 illustrates a case in which a training sequence is replicated 4 times. Two possible positions to start the synchronization algorithm are displayed, and both of them corresponds exactly the same correlation output.

[0081] It must still be determined which one has been detected, in order to provide the value of $n_{synchro}$. However, because any of the replicas of the training sequence are identical, there is no way to determine if the first, the second or the last replica has been detected by the cross-correlation step. Although the detection step which will be now described is independent of the type of correlation used, either sliding or slice-splitting correlation, it is assumed now that the cross-correlation means are adapted to perform a slice-splitting correlation.

[0082] To overcome the problem of ambiguity modulo N samples, an additional replica of the training sequence is inserted just after the $M+1$ replicas, but flipped. This differentiates it from the other replicas. Figure

12 illustrates the procedure in which LFR designates the last flipped replica of the training sequence TS1. All in all, the number of replicas of the training sequence in the beacon preamble is $Q = M + 2$.

[0083] As indicated above, the detection step according to the invention comprises in fact a first sub-step of detecting one replica of the training sequence, as well as a second sub-step alignment phase including sequentially scanning the next replicas until the flipped last one is found. In this respect, detection means DM (Figure 5) comprises scanning means SCM. More precisely, the scanning means comprises correlation means CMX which are adapted, in this example, to perform also a slice-splitting correlation, and comparison means CMPX as well as control means CLTM.

[0084] The flow chart of this alignment phase of the detection step, including also a slice-splitting correlation 131, is illustrated in Figure 13. Let the detected replica be the i^{th} out of Q . The receiver has no knowledge of which is the actual value of i ($i = 1, 2, \dots, Q$). The only information in its possession is the position of the peak n_{peak} and its sign α_{peak} , relatively to the i^{th} replica of the training sequence. For $M = 1$ (no division of the correlation in slices), n_{peak} can assume values from 0 up to N while if $M > 1$, n_{peak} can assume values from 0 to N/M . It can be verified that in both cases n_{peak} corresponds to the number of samples that separate the reference point n_m assumed in the correlation process and the beginning of the $(i+1)^{\text{th}}$ replica of training sequence. In this way the receiver is able to align with the beginning of the $(i+1)^{\text{th}}$ replica, $(i+2)^{\text{th}}$, $(i+k)^{\text{th}}$ replica and so on, as all the

replicas are N samples long.

[0085] The receiver can now evaluate the correlation between the $(i+1)^{\text{th}}$ replica and the training sequence. In this step it is not necessary to compute a large portion of the full correlation (over a window of N or N/M samples), as the position of the peak is known from the previous cross-correlation step. Therefore, only a zero-shift correlation is evaluated, by simply adding the samples in the positions prescribed by the pseudo-random code.

$$z^{(k)} = \sum_{j=0}^{L-1} a_j r_{n_m + n_{peak} + c_j + kN}, \quad k = 1, 2, \dots \quad (16)$$

[0086] The result z of the process for $k = 1$ is then checked against a threshold $TH2$. If the absolute value of z is below the threshold $TH2$, a false alarm in the first step of the algorithm is assumed to have taken place and the cross-correlation step starts over. If the absolute value of z exceeds the threshold $TH2$, two cases are possible: if the sign of z is equal to α_{peak} , the receiver aligns with the $(i+2)^{\text{th}}$ replica of the training sequence and applies again the alignment step of the algorithm. On the other hand, if the sign of z is different than α_{peak} , the receiver becomes aware of being aligned with the Q^{th} and last replica of the training sequence inside the beacon preamble and is then able to synchronize with the end of it.

[0087] After the cross-correlation step of the algorithm there is a fair certainty that the detected peak was not due to a false alarm. Moreover the alignment step of the algorithm is to be performed only

a limited number of times, and thus false alarms are not likely to happen. Hence, in the alignment step of the algorithm a lower threshold TH2 can be used. This new decision affects only slightly the overall probability of missed detection and false alarm, in a way that is easy to compute. Empirically the threshold TH2 to be used in the alignment step could be such that the probability of false alarm in the decision is on the order of 10^{-4} , namely on the same order of magnitude of the probability of false alarm in the cross-correlation step of the algorithm evaluated over a super frame duration.

[0088] As it has been determined, during the slice splitting correlation steps, when a peak exceeds the threshold, it is not likely to be the only one. If multipath propagation is assumed, after the correlation many paths could be detected, each corresponding to a $(n_{peak}, \alpha_{peak})$ pair. This is valuable information that can be exploited to increase the performance of the alignment step of the algorithm. In fact, instead of considering only a position for the alignment procedure, the detected paths can be combined in a rake-like fashion. If the number of detected peaks is U , the alignment step of the algorithm computes the correlation value z in this way:

$$z^{(k)} = \frac{1}{U} \sum_{u=1}^U \alpha_{peak}^{(u)} \sum_{j=0}^{L-1} a_j r_{n_m + n_{peak}^{(u)} + c_j + kN}, \quad k = 1, 2, \dots \quad (17)$$

[0089] The operation corresponds to a coherent correlation with a rough estimate of the channel impulse response, namely a distance metric calculation.

It is to be noted that the detected paths are not combined according to their estimated amplitude, but are only multiplied by their sign. This is a sub-optimal approach but has the advantage of a lower complexity, with only a slight loss of performance.

[0090] In the multipath case it is not possible to compare the correlation output with the sign of a single peak. In a more correct way the sign of z should be compared with the result of the same correlation operator in Equation (17) on the window of the received signal that originated the detection, namely with the sign of:

$$z^{(0)} = \frac{1}{U} \sum_{u=1}^U \alpha_{peak}^{(u)} \sum_{j=0}^{L-1} a_j r_{n_m + n_{peak}^{(u)} + c_j} \quad (18)$$

In other words, replicas of the training sequence (convolved with the multipath channel) are seen as a series of binary symbols (either standard or flipped). The object of the alignment is to demodulate these binary symbols and especially correctly find the last one, which is the only one with opposite polarity. The aggregated energy of different paths enables to use an even lower threshold in the alignment step.

[0091] As illustrated now in Figure 14, the beacon preamble includes at least $M+1$ synchronization slots SS_i corresponding respectively to the replicas of the training sequence. If the signal-to-noise ratio is high enough, each slice can be computed using one synchronization slot.

[0092] However, in case the actual operating signal to noise ratio is low, one synchronization slot SS may

not be sufficient to gather enough energy for taking a decision with the desired degree of accuracy, and therefore, a longer period of time is required. Without changing the synchronization slot structure it is possible to achieve a better performance, by pairing adjacent synchronization slots SS, and using both of them to test a single slice (Figure 15). The energy of the two synchronization slots SS is added, and only after the second synchronization slot SS a decision is taken. In this case, during a period of duration MT_{train} (corresponding to M synchronization slots SS), it is possible to test only $M/2$ slices, e.g., the first half. Equation (15) is changed into:

$$x_n^{(m)} = \sum_{u=0}^1 \sum_{j=0}^{L-1} a_j r_{n+m \left(2N + \frac{N}{M} \right) + uN + c_j} \quad (19)$$

[0093] The second half of the slices could be tested in the next M synchronization slots SS. Yet, if the second half of the slices is tested straight afterwards, there could be a missed detection, as during this period there could be a beacon preamble whose correct synchronization is located in the first $M/2$ slices. Therefore, after the round of M synchronization slots SS, it is necessary to test again the first $M/2$ slices. Only after a whole super frame period the last $M/2$ can be tested, throughout the next super frame duration. The worst-case synchronization time is then the duration of two super frames.

[0094] The invention is not limited to the above described embodiments. More precisely, although the sliding correlation or the slice splitting correlation

have been described for performing a coarse synchronization between the coordinator PNC and an independent data device DEV, the correlations according to the invention can also be used for performing a frame synchronization between two independent data devices communicating with each other, using the training sequence TS2 contained in the preamble PRB of the frame Fr_i (Figure 2).

[0095] Further, the training sequence can be known by the device, or eventually unknown but belonging to a set of possible training sequences. In such a case, all the training sequences can be successfully tested using the synchronization method accordingly to the invention. The actual training sequence would be, for example, the one that leads effectively to a peak of detection. And the feature of flipped last replica can be extended to a pattern of flipped/unflipped training sequence replicas following, for instance, a Barker code.

[0096] Further, the above description is an example of application to a particular network structure with a particular super frame structure. However, the invention is not limited to this system. More generally, the slice splitting correlation of the invention can be used for performing a cross-correlation of a received signal on blocks of N samples when the size of the receiving buffer is smaller than N .

[0097] In other words, the invention proposes also a method for performing a digital cross-correlation of an incident signal containing a periodic training sequence, with the training sequence. The training sequence has $M+1$ replicas, each replica having a size

of N samples and containing L pulses, with M being a sub-multiple of N greater than or equal to 2. The digital cross-correlation is performed iteratively in a block-by-block fashion. The computation of each block is split into M slices which are computed by algebraically summing windows N/M samples long.